

## AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings of claims in the application:

### LISTING OF CLAIMS

1. (Currently Amended) A small footprint device comprising:
- at least one processing element configured to execute groups of one or more program modules in separate contexts, said one or more program modules comprising zero or more sets of executable instructions and zero or more sets of data definitions, said zero or more sets of executable instructions and said zero or more data definitions grouped as object definitions, each context comprising a protected object instance space such that at least one of said object definitions is instantiated in association with a particular context ~~objects of a program module associated with a particular context;~~
- a memory comprising instances of objects; and
- a context barrier for separating and isolating said contexts, said context barrier configured to ~~use said memory to control object-oriented access of a program module executing in one context to information and/or a program module executing in another context~~ for controlling execution of at least one instruction of one of said zero or more sets of instructions comprised by a program module based at least in part on whether said at least one instruction is executed for an object instance associated with a first one of said one or more separate contexts and whether said at least one instruction is requesting access to an instance of an

object definition associated with a second one of said one or more separate contexts, said context barrier further configured to prevent said access if said access is unauthorized and enable said access if said access is authorized.

2-24 (Cancelled)

25. (Previously Presented) The small footprint device of claim 1 in which said at least one processing element is a virtual machine running on a processor.

26. (Previously Presented) The small footprint device of claim 25 in which said virtual machine runs on top of a card operating system.

27. (Cancelled)

28. (Cancelled)

29. (Previously Presented) The small footprint device of claim 1 in which said context barrier allocates separate respective name spaces for each context.

30. (Previously Presented) The small footprint device of claim 1 in which said context barrier allocates separate respective memory spaces for each context.

31. (Previously Presented) The small footprint device of claim 1 in which at least one program module comprises a plurality of applets.
32. (Previously Presented) The small footprint device of claim 1 in which said context barrier enforces at least one security check on at least one of principal, object or action to prevent access from a principal in one context to an object in a different context.
33. (Cancelled)
34. (Cancelled)
35. (Previously Presented) The small footprint device of claim 32 in which at least one security check is based on partial name agreement between a principal and an object.
36. (Previously Presented) The small footprint device of claim 32 in which at least one security check is based on memory space agreement between a principal and an object.
37. (Currently Amended) A method of operating a small footprint device that includes a processing machine, wherein program modules are executed on the processing machine, the method comprising:

executing groups of one or more program modules in separate contexts, said one or more program modules comprising zero or more sets of executable instructions and zero or more sets of data definitions, said zero or more sets of executable instructions and said zero or more data definitions grouped as object definitions, each context comprising a protected object instance space such that at least one of said object definitions is instantiated in association with a particular context ~~objects of a program module associated with a particular context~~; and providing a context barrier for separating and isolating said contexts and for ~~controlling the object oriented access of a program module executing in one context to information and/or a program module executing in another context~~ controlling execution of at least one instruction of one of said zero or more sets of instructions comprised by a program module based at least in part on whether said at least one instruction is executed for an object instance associated with a first one of said one or more separate contexts and whether said at least one instruction is requesting access to an instance of an object definition associated with a second one of said one or more separate contexts, said providing further comprising:

preventing said access if said access is unauthorized; and


enabling said access if said access is authorized.

38. (Previously Presented) The method of claim 37 in which the context barrier is implemented using a virtual machine.

39. (Cancelled)

40. (Cancelled)

41. (Previously Presented) The method of claim 37 in which the context barrier will not permit a principal to access an object unless both principal and object are part of the same name space.



42. (Previously Presented) The method of claim 37 in which the context barrier will not permit a principal to access an object unless both principal and object are part of the same memory space.

43. (Previously Presented) The method of claim 37 in which the context barrier will not permit a principal to perform an action on an object unless both principal and object are part of the same context.

44. (Previously Presented) The method of claim 43 in which the context barrier will permit a principal to perform an action on an object when they are not part of the same context if the principal is authorized to perform the action on the object.

45. (Previously Presented) The method of claim 44 in which the principal is authorized if it passes at least one security check.

46. (Previously Presented) The method of claim 45 in which said at least one security check is one of a plurality of security checks.
47. (Previously Presented) The method of claim 44 in which, if a principal in a first context is authorized to perform one or more actions on an object in a second context, when the action is performed it will execute within the second context.
48. (Previously Presented) The method of claim 47 in which, when one or more actions are authorized in the second context, subsequent actions will be authorized based on executing in the second context, and a principal in the second context will be able to access objects in the second context.
49. (Previously Presented) The method of claim 48 in which, when one or more actions complete in the second context, execution will return to the first context.
50. (Previously Presented) The method of claim 47 in which, when action is undertaken in the second context that requires access to an object in a third context, the action will execute within the third context.
51. (Previously Presented) The method of claim 50 in which switches to a new context will occur any time action is authorized on an object in a new context.
52. (Currently Amended) A computer program product, comprising:

a memory medium; and

a computer controlling element comprising instructions for implementing a context

barrier on a small footprint device, said small footprint device comprising:

at least one processing element configured to execute groups of one or more program

modules in separate contexts, said one or more program modules comprising

zero or more sets of executable instructions and zero or more sets of data

definitions, said zero or more sets of executable instructions and said zero or

more data definitions grouped as object definitions, each context comprising a

protected object instance space such that at least one of said object definitions is

instantiated in association with a particular context ~~objects of a program module~~

~~associated with a particular context;~~

a memory comprising instances of objects; and

a context barrier for separating and isolating said contexts, said context barrier

~~configured to use said memory to control object-oriented access of a program~~

~~module executing in one context to information and/or a program module~~

~~executing in another context~~ for controlling execution of at least one instruction

of one of said zero or more sets of instructions comprised by a program module

based at least in part on whether said at least one instruction is executed for an

object instance associated with a first one of said one or more separate contexts

and whether said at least one instruction is requesting access to an instance of an

object definition associated with a second one of said one or more separate

contexts, said context barrier further configured to prevent said access if said

access is unauthorized and enable said access if said access is authorized.

53. (Previously Presented) The computer program product of claim 52 in which said memory medium is a carrier wave.

54. (Currently Amended) A computer program product, comprising:

a memory medium; and

a computer controlling element comprising instructions for separating a plurality of

programs on a small footprint device, said small footprint device comprising:

at least one processing element configured to execute groups of one or more

program modules in separate contexts, said one or more program modules

comprising zero or more sets of executable instructions and zero or more

sets of data definitions, said zero or more sets of executable instructions and

said zero or more data definitions grouped as object definitions, each context

comprising a protected object instance space such that at least one of said

object definitions is instantiated in association with a particular context

objects of a program module associated with a particular context;

a memory comprising instances of objects; and

a context barrier for separating and isolating said contexts, said context barrier

~~configured to use said memory to control object-oriented access of a~~

~~program module executing in one context to information and/or a program~~

~~module executing in another context~~ for controlling execution of at least one


instruction of one of said zero or more sets of instructions comprised by a

program module based at least in part on whether said at least one



instruction is executed for an object instance associated with a first one of  
said one or more separate contexts and whether said at least one instruction  
is requesting access to an instance of an object definition associated with a  
second one of said one or more separate contexts, said context barrier further  
configured to prevent said access if said access is unauthorized and enable  
said access if said access is authorized.

55. (Previously Presented) The computer program product of claim 54 in which said  
memory medium is a carrier wave.



56. (Currently Amended) A carrier wave carrying instructions for implementing a  
context barrier on a small footprint device over a communications link, said small  
footprint device comprising:  
at least one processing element configured to execute groups of one or more program  
modules in separate contexts, said one or more program modules comprising  
zero or more sets of executable instructions and zero or more sets of data  
definitions, said zero or more sets of executable instructions and said zero or  
more data definitions grouped as object definitions, each context comprising a  
protected object instance space such that at least one of said object definitions is  
instantiated in association with a particular context ~~objects of a program module~~  
~~associated with a particular context~~;  
a memory comprising instances of objects; and

2

a context barrier for separating and isolating said contexts, said context barrier configured to use said memory to control object-oriented access of a program module executing in one context to information and/or a program module executing in another context for controlling execution of at least one instruction of one of said zero or more sets of instructions comprised by a program module based at least in part on whether said at least one instruction is executed for an object instance associated with a first one of said one or more separate contexts and whether said at least one instruction is requesting access to an instance of an object definition associated with a second one of said one or more separate contexts, said context barrier further configured to prevent said access if said access is unauthorized and enable said access if said access is authorized.

57. (Currently Amended) A carrier wave carrying instructions over a communications link for separating a plurality of programs on a small footprint device, said small footprint device comprising:
- at least one processing element configured to execute groups of one or more program modules in separate contexts, said one or more program modules comprising zero or more sets of executable instructions and zero or more sets of data definitions, said zero or more sets of executable instructions and said zero or more data definitions grouped as object definitions, each context comprising a protected object instance space such that at least one of said object definitions is instantiated in association with a particular context ~~objects of a program module associated with a particular context;~~

a memory comprising instances of objects; and

a context barrier for separating and isolating said contexts, said context barrier configured ~~to use said memory to control object oriented access of a program module executing in one context to information and/or a program module executing in another context~~ for controlling execution of at least one instruction of one of said zero or more sets of instructions comprised by a program module based at least in part on whether said at least one instruction is executed for an object instance associated with a first one of said one or more separate contexts and whether said at least one instruction is requesting access to an instance of an object definition associated with a second one of said one or more separate contexts, said context barrier further configured to prevent said access if said access is unauthorized and enable said access if said access is authorized.

- 2
58. (Currently Amended) A method of shipping code over a network, comprising transmitting a block of code from a server, said block of code comprising instructions over a communications link for separating a plurality of programs on a small footprint device, said small footprint device comprising:
- at least one processing element configured to execute groups of one or more program modules in separate contexts, said one or more program modules comprising zero or more sets of executable instructions and zero or more sets of data definitions, said zero or more sets of executable instructions and said zero or more data definitions grouped as object definitions, each context comprising a protected object instance space such that at least one of said object definitions is

instantiated in association with a particular context ~~objects of a program module~~  
~~associated with a particular context;~~

a memory comprising instances of objects; and

a context barrier for separating and isolating said contexts, said context barrier  
configured to ~~use said memory to control object oriented access of a program~~  
~~module executing in one context to information and/or a program module~~  
~~executing in another context~~ for controlling execution of at least one instruction  
of one of said zero or more sets of instructions comprised by a program module  
based at least in part on whether said at least one instruction is executed for an  
object instance associated with a first one of said one or more separate contexts  
and whether said at least one instruction is requesting access to an instance of an  
object definition associated with a second one of said one or more separate  
contexts, said context barrier further configured to prevent said access if said  
access is unauthorized and enable said access if said access is authorized.

59. (New) The small footprint device of claim 1 wherein an object instance is associated  
with a context by recording the name of said context in a header of said object  
instance, information in said header inaccessible to said one or more program  
modules.

60. (New) The small footprint device of claim 1 wherein  
said memory comprises object header data, said object header data comprising  
information associated with at least one of said instances of objects; and

said controlling execution is based at least in part on said object header data.

61. (New) The small footprint device of claim 1 wherein

said memory is partitioned into a plurality of memory spaces with instances of  
objects allocated for storage in one of said plurality of storage spaces; and  
said controlling execution is based at least in part on determining the storage space  
allocated to an executing object instance and an accessed object instance.

62. (New) The method of claim 37 wherein an object instance is associated with a context  
by recording the name of said context in a header of said object instance, information  
in said header inaccessible to said one or more program modules.

63. (New) The method of claim 37 wherein said controlling execution is based at least in  
part on object header data comprising information associated with at least one of said  
instances of objects.

64. (New) The method of claim 37 wherein

a memory of said small footprint device is partitioned into a plurality of memory  
spaces with instances of objects allocated for storage in one of said plurality of  
storage spaces; and  
said controlling execution is based at least in part on determining the storage space  
allocated to an executing object instance and an accessed object instance.